

# Degree Bounded Matroids and Submodular Flows

Tamás Király<sup>1\*</sup>, Lap Chi Lau<sup>2</sup>, and Mohit Singh<sup>3\*\*</sup>

<sup>1</sup> MTA-ELTE Egerváry Research Group, Dept. of Operations Research, Eötvös Loránd University, Budapest. [tkiraly@cs.elte.hu](mailto:tkiraly@cs.elte.hu)

<sup>2</sup> Dept. of Computer Science and Engineering, The Chinese University of Hong Kong. [chi@cse.cuhk.edu.hk](mailto:chi@cse.cuhk.edu.hk)

<sup>3</sup> Tepper School of Business, Carnegie Mellon University. [mohits@andrew.cmu.edu](mailto:mohits@andrew.cmu.edu)

**Abstract.** We consider two related problems, the MINIMUM BOUNDED DEGREE MATROID BASIS problem and the MINIMUM BOUNDED DEGREE SUBMODULAR FLOW problem. The first problem is a generalization of the MINIMUM BOUNDED DEGREE SPANNING TREE problem: we are given a matroid and a hypergraph on its ground set with lower and upper bounds  $f(e) \leq g(e)$  for each hyperedge  $e$ . The task is to find a minimum cost basis which contains at least  $f(e)$  and at most  $g(e)$  elements from each hyperedge  $e$ . In the second problem we have a submodular flow problem, a lower bound  $f(v)$  and an upper bound  $g(v)$  for each node  $v$ , and the task is to find a minimum cost 0-1 submodular flow with the additional constraint that the sum of the incoming and outgoing flow at each node  $v$  is between  $f(v)$  and  $g(v)$ . Both of these problems are NP-hard (even the feasibility problems are NP-complete), but we show that they can be approximated in the following sense. Let  $\text{OPT}$  be the value of the optimal solution. For the first problem we give an algorithm that finds a basis  $B$  of cost no more than  $\text{OPT}$  such that  $f(e) - 2\Delta + 1 \leq |B \cap e| \leq g(e) + 2\Delta - 1$  for every hyperedge  $e$ , where  $\Delta$  is the maximum degree of the hypergraph. If there are only upper bounds (or only lower bounds), then the violation can be decreased to  $\Delta - 1$ . For the second problem we can find a 0-1 submodular flow of cost at most  $\text{OPT}$  where the sum of the incoming and outgoing flow at each node  $v$  is between  $f(v) - 1$  and  $g(v) + 1$ . These results can be applied to obtain approximation algorithms for different combinatorial optimization problems with degree constraints, including the MINIMUM CROSSING SPANNING TREE problem, the MINIMUM BOUNDED DEGREE SPANNING TREE UNION problem, the MINIMUM BOUNDED DEGREE DIRECTED CUT COVER problem, and the MINIMUM BOUNDED DEGREE GRAPH ORIENTATION problem.

## 1 Introduction

In this paper we consider combinatorial optimization problems with degree constraints, for which the corresponding feasibility problem is already NP-complete.

---

\* Research supported by OTKA K60802 and OMFB-01608/2006.

\*\* Research supported by NSF grant CCF-0728841.

One approach to deal with these problems is to allow a slight violation of the degree constraints, and find a solution of this relaxation that has small cost. A prime example of this approach is the MINIMUM BOUNDED DEGREE SPANNING TREE problem, where we have upper (and possibly lower) bounds on the degree of the spanning tree at each node. The corresponding feasibility problem is NP-complete since it includes the Hamiltonian path problem. Goemans [8] showed that if the value of the optimal solution is  $\text{OPT}$ , then one can find in polynomial time a spanning tree of cost at most  $\text{OPT}$  that violates the degree bounds by at most 2. Using the iterative relaxation method, which is also the main technique in the present paper, Singh and Lau [12] gave an algorithm that finds a spanning tree of cost at most  $\text{OPT}$  that violates the bounds by at most 1. The aim of this paper is to obtain similar results for more general combinatorial optimization problems.

### 1.1 Minimum Bounded Degree Matroid Basis

The first problem considered is the MINIMUM BOUNDED DEGREE MATROID BASIS problem, which is a generalization of the MINIMUM BOUNDED DEGREE SPANNING TREE problem. We are given a matroid  $M = (V, \mathcal{I})$ , a cost function  $c : V \rightarrow \mathbb{R}$ , a hypergraph  $H = (V, E)$ , and lower and upper bounds  $f(e)$  and  $g(e)$  for each hyperedge  $e \in E(H)$ . The task is to find a basis  $B$  of minimum cost such that  $f(e) \leq |B \cap e| \leq g(e)$  for each hyperedge  $e \in E(H)$ . One motivation for considering the matroid generalization was the following problem posed by Frieze [7]: “Given a binary matroid  $M_A$  over the columns of a 0,1-matrix  $A$  and bounds  $g_i$  for each row  $i$  of  $A$ , find a basis  $B$  of matroid  $M_A$  such that there are at most  $g_i$  ones in any row among columns in  $B$ ”.

A problem similar to ours has been considered recently by Chaudhuri et al. [3]. The results we give in this paper improve considerably their approximation guarantees. Our first main result is the following:

**Theorem 1.** *There exists a polynomial time algorithm for the MINIMUM BOUNDED DEGREE MATROID BASIS problem which returns a basis  $B$  of cost at most  $\text{OPT}$  such that  $f(e) - 2\Delta + 1 \leq |B \cap e| \leq g(e) + 2\Delta - 1$  for each  $e \in E(H)$ . Here  $\Delta = \max_{v \in V} |\{e \in E(H) : v \in e\}|$  is the maximum degree of the hypergraph  $H$  and  $\text{OPT}$  is the cost of an optimal solution which satisfies all the degree constraints.*

This theorem can be improved if only upper bounds (or only lower bounds) are present. The proof of the improvement uses the proof technique of Bansal et al. [1], who worked independently on the MINIMUM CROSSING SPANNING TREE problem and obtained the following result for that special case.

**Theorem 2.** *There exists a polynomial time algorithm for the MINIMUM BOUNDED DEGREE MATROID BASIS problem with only upper bounds which returns a basis  $B$  of cost at most  $\text{OPT}$  such that  $|B \cap e| \leq g(e) + \Delta - 1$  for each  $e \in E(H)$ . An analogous result holds when only lower bounds are present.*

It should be noted that this does not match the result of Singh and Lau [12] on minimum bounded degree spanning trees, since that result violates the degree bounds by at most 1 even when both upper and lower bounds are present. Somewhat surprisingly, we show an example at the end of Section 3 indicating that obtaining a result for general matroids which satisfies both upper and lower degree bounds within additive error of one does not follow from current techniques.

## 1.2 Minimum Bounded Degree Submodular Flow

The second problem considered in this paper is the MINIMUM BOUNDED DEGREE SUBMODULAR FLOW problem. Here we are given a digraph  $D = (V, E)$ , a crossing submodular set function  $b : 2^V \rightarrow \mathbb{Z} \cup \{+\infty\}$ , node sets  $V_f \subseteq V$  and  $V_g \subseteq V$ , and functions  $f : V_f \rightarrow \mathbb{Z}_+$  and  $g : V_g \rightarrow \mathbb{Z}_+$ . Let us introduce the following notation for the set of edges entering or leaving a node set:

$$\begin{aligned}\delta^{in}(X) &= \{uv \in E : u \notin X, v \in X\}, \\ \delta^{out}(X) &= \{uv \in E : u \in X, v \notin X\}, \\ \delta(X) &= \delta^{in}(X) \cup \delta^{out}(X).\end{aligned}$$

If  $F \subseteq E$  is an edge set and  $x : E \rightarrow \mathbb{R}$  is a function on the edges, then we use the notation  $x(F) = \sum_{e \in F} x(e)$ . A *degree-constrained 0-1 submodular flow* is a vector  $x \in E \rightarrow \{0, 1\}$  with the following properties:

$$x(\delta^{in}(X)) - x(\delta^{out}(X)) \leq b(X) \quad \text{for every } X \subseteq V, \quad (1)$$

$$x(\delta(v)) \geq f(v) \quad \text{for every } v \in V_f, \quad (2)$$

$$x(\delta(v)) \leq g(v) \quad \text{for every } v \in V_g. \quad (3)$$

If  $V_f = V_g = \emptyset$ , then this is the well-studied submodular flow problem, introduced by Edmonds and Giles [5]. There are several efficient algorithms for finding a feasible submodular flow, or even a minimum cost submodular flow for a linear cost function. However, the addition of the degree constraints (2) and (3) makes the feasibility problem NP-complete, as we show in subsection 4.1. Our second main result is the following:

**Theorem 3.** *There exists a polynomial time algorithm for the MINIMUM BOUNDED DEGREE SUBMODULAR FLOW problem which returns a 0-1 submodular flow of cost at most OPT that violates each degree constraint by at most one, where OPT is the cost of an optimal solution which satisfies all the degree constraints.*

In Section 2, we show some applications of the main results. Then we present the proofs of the main results and some corresponding hardness results in Section 3 for the matroid problem and in Section 4 for the submodular flow problem.

## 2 Applications

In this section we highlight some applications of the main results.

## 2.1 Minimum Crossing Spanning Tree

In the MINIMUM CROSSING SPANNING TREE problem, we are given a graph  $G = (V, E)$  with edge cost function  $c$ , a collection of cuts (edge subsets)  $\mathcal{C} = \{C_1, \dots, C_m\}$  and bound  $g_i$  for each cut  $C_i$ . The task is to find a tree  $T$  of minimum cost such that  $T$  contains at most  $g_i$  edges from cut  $C_i$ . See [2] for various applications of this problem. The MINIMUM BOUNDED DEGREE SPANNING TREE problem is the special case where  $\mathcal{C} = \{\delta(v) : v \in V\}$ . The following result<sup>4</sup> (see also [1]) can be obtained as a corollary of Theorem 2. Note that  $d = 2$  for the MINIMUM BOUNDED DEGREE SPANNING TREE problem.

**Corollary 1.** [1] *There exists a polynomial time algorithm for the MINIMUM CROSSING SPANNING TREE problem that returns a tree  $T$  with cost at most  $\text{OPT}$  and such that  $T$  contains at most  $g_i + d - 1$  edges from cut  $C_i$  for each  $i$  where  $d = \max_{e \in E} |\{C_i : e \in C_i\}|$ . Here  $\text{OPT}$  is the cost of an optimal solution which satisfies all the cut constraints.*

*Proof.* Let  $M = (E, \mathcal{I})$  denote the graphic matroid over the graph  $G$ . The hypergraph  $H$  is defined with  $V(H) = E(G)$  and  $E(H) = \{C_i : 1 \leq i \leq m\}$ . Note that  $\Delta = \max_{v \in V(H)} |\{e \in E(H) : v \in e\}| = \max_{e \in E(G)} |\{C_i : e \in C_i\}| = d$ . So, using Theorem 2, we obtain a basis  $T$  of matroid  $M$  (which is a spanning tree), such that  $|T \cap C_i| \leq g_i + d - 1$ .  $\square$

## 2.2 Minimum Bounded-Ones Binary Matroid Basis

For the MINIMUM BOUNDED-ONES BINARY MATROID BASIS problem posted by Frieze [7], we are given a binary matroid  $M_A$  over the columns of a 0, 1-matrix  $A$  and bounds  $g_i$  for each row  $i$  of  $A$ . The task is to find a minimum cost basis  $B$  of matroid  $M_A$  such that there are at most  $g_i$  ones in any row among columns in  $B$ . The following result is obtained as a corollary of Theorem 2.

**Corollary 2.** *There exists a polynomial time algorithm for the MINIMUM BOUNDED-ONES BINARY MATROID BASIS problem which returns a basis  $B$  of cost at most  $\text{OPT}$  such that there are at most  $g_i + d - 1$  ones in any row restricted to columns of  $B$ . Here  $d$  is the maximum number of ones in any column of  $A$  and  $\text{OPT}$  is the cost of an optimal solution satisfying all the row constraints.*

*Proof.* Let  $M = M_A$  and define a hypergraph  $H$  where the vertex set is the columns of  $A$ . The hyperedges correspond to rows of  $A$  where  $e_i = \{A^j : A_{ij} = 1\}$  where  $A^j$  is the  $j^{\text{th}}$  column of  $A$ . Note that  $\Delta = \max_{v \in V(H)} |\{e \in E(H) : v \in e\}| = \max_j |\{i : A_{ij} = 1\}| = d$ , which is the maximum number of ones in any column of  $A$ . So, using Theorem 2, we obtain a basis of  $M = M_A$  such that number of ones in any row is at most  $g_i + d - 1$ .  $\square$

<sup>4</sup> Independent of the work in [1], we obtained Corollary 1 with a weaker bound using Theorem 1.

### 2.3 Minimum Bounded Degree Spanning Tree Union

In the MINIMUM BOUNDED DEGREE SPANNING TREE UNION problem, we are given a graph  $G = (V, E)$  with edge cost function  $c$ , a positive integer  $k$ , and lower and upper degree bounds  $f(v)$  and  $g(v)$  for each vertex  $v$ . The task is to find a subgraph  $H$  which is the union of  $k$  edge-disjoint spanning trees and the degree of  $v$  in  $H$  is between  $f(v)$  and  $g(v)$ . The MINIMUM BOUNDED DEGREE SPANNING TREE problem is a special case when  $k = 1$ . Theorem 2 implies the following result, which is optimal in terms of the degree upper bounds.

**Corollary 3.** *There exists a polynomial time algorithm for the MINIMUM BOUNDED DEGREE SPANNING TREE UNION problem which returns a subgraph  $G$  of cost at most  $\text{OPT}$  which is the union of  $k$  edge-disjoint spanning trees and the degree of  $v$  in  $H$  is at most  $g(v) + 1$ . Here  $\text{OPT}$  is the cost of an optimal solution which satisfies all the degree upper bounds.*

*Proof.* Let  $M = (E, \mathcal{I})$  denote the union of  $k$  graphic matroids over the graph  $G$ , which is a matroid by the matroid union theorem. The hypergraph  $H$  is defined with  $V(H) = E(G)$  and  $E(H) = \{\delta(v) : v \in V(G)\}$ . Note that  $\Delta = \max_{v \in V(H)} |\{e \in E(H) : v \in e\}| = \max_{e \in E(G)} |\{\delta(v) : v \in V(G) \wedge e \in \delta(v)\}| = 2$ . So, using Theorem 2, we obtain a basis  $T$  of matroid  $M$  (which is the union of  $k$  edge-disjoint spanning trees), such that  $|T \cap C_i| \leq g_i + 1$ .  $\square$

### 2.4 Minimum Bounded Degree Directed Cut Cover

Let  $D = (V, E)$  be a digraph. A set of vertices  $X$  is called a *directed cut* if  $\delta^{\text{out}}(X) = \emptyset$ . A subset of edges  $F$  is called a *directed cut cover* if  $|F \cap \delta(X)| \neq \emptyset$  for every directed cut  $X$ . In the MINIMUM BOUNDED DEGREE DIRECTED CUT COVER problem, we are given a digraph  $D = (V, E)$ , a cost function  $c : E \rightarrow \mathbb{Z}$ , and degree constraints  $f(v)$  and  $g(v)$  for each  $v \in V$ . The task is to find a directed cut cover  $F \subseteq E$  of minimum cost such that  $f(v) \leq |F \cap \delta(v)| \leq g(v)$  for every  $v \in V$ . Theorem 3 implies the following result, which is optimal in terms of the degree bounds.

**Corollary 4.** *There exists a polynomial time algorithm for the MINIMUM BOUNDED DEGREE DIRECTED CUT COVER problem which returns a directed cut cover  $F$  of cost at most  $\text{OPT}$  and  $f(v) - 1 \leq |F \cap \delta(v)| \leq g(v) + 1$  for each vertex  $v \in V$ . Here  $\text{OPT}$  is the cost of an optimal solution which satisfies all the degree constraints.*

*Proof.* Let  $b(X) = -1$  if  $V \setminus X$  is a directed cut, and let  $b(X) = \infty$  otherwise. Then  $b$  is a crossing submodular set function. In this setting, a 0-1 submodular flow corresponds to a directed cut cover. So, by Theorem 3, we obtain a directed cut cover  $F$  such that  $f(v) - 1 \leq |F \cap \delta(v)| \leq g(v) + 1$  for every  $v \in V$ .  $\square$

## 2.5 Minimum Bounded Degree Graph Orientation

In the MINIMUM BOUNDED DEGREE GRAPH ORIENTATION problem, we are given a digraph  $D = (V, E)$ , a cost function  $c : E \rightarrow \mathbb{Z}$ , and bounds  $f(v) \leq g(v)$  for every  $v \in V$ . The task is to find an edge set of minimum cost whose reversal makes the digraph  $k$ -edge-connected, so that the number of edges reversed at each node  $v$  is between  $f(v)$  and  $g(v)$ . Theorem 3 implies the following result, which is optimal in terms of the degree bounds.

**Corollary 5.** *There exists a polynomial time algorithm for the MINIMUM BOUNDED DEGREE GRAPH ORIENTATION problem which finds an edge set of cost at most  $\text{OPT}$  whose reversal makes the digraph  $k$ -edge-connected and such that the number of edges reversed at each node  $v$  is between  $f(v) - 1$  and  $g(v) + 1$ . Here  $\text{OPT}$  is the cost of an optimal solution which satisfies all the degree constraints.*

*Proof.* This can be done by considering the submodular flow problem defined by the set function  $b(X) = |\delta^{\text{in}}(X)| - k$  ( $\emptyset \neq X \subsetneq V$ ) (see [6]), which is a submodular set function. In this setting, a 0-1 submodular flow corresponds to an edge set whose reversal makes the digraph strongly  $k$ -edge-connected. So this result follows from Theorem 3.  $\square$

It is shown in subsection 4.1 that the corresponding feasibility problem is NP-complete, and thus the feasibility problem for bounded degree submodular flow is also NP-complete.

## 3 Minimum Bounded Degree Matroid Basis

**Proof of Theorem 1:** The main technique used to prove Theorem 1 is the iterative relaxation method used in [10, 12], which is based on the iterative rounding method introduced by Jain [9]. We first formulate a linear programming relaxation for the MINIMUM DEGREE BOUNDED MATROID BASIS problem. Let  $r : 2^V \rightarrow \mathbb{Z}_+$  denote the rank function of matroid  $M$ .

$$\begin{array}{ll} \text{minimize} & c(x) = \sum_{v \in V} c_v x_v \end{array} \quad (4)$$

$$\text{subject to} \quad x(V) = r(V) \quad (5)$$

$$x(S) \leq r(S) \quad \forall S \subseteq V \quad (6)$$

$$f(e) \leq x(e) \leq g(e) \quad \forall e \in E(H) \quad (7)$$

$$0 \leq x_v \leq 1 \quad \forall v \in V \quad (8)$$

This linear program is exponential in size but can be separated over in polynomial time if given an access to the independent set oracle [4]. Given a matroid  $M = (V, \mathcal{I})$  and an element  $v \in V$ , we denote by  $M \setminus v$  the matroid obtained by deleting  $v$ , i.e.,  $M \setminus v = (V', \mathcal{I}')$  where  $V' = V \setminus \{v\}$  and  $\mathcal{I}' = \{S \in \mathcal{I} : v \notin S\}$ . We

1. Initialization  $B \leftarrow \emptyset$ ,
2. While  $B$  is not a basis do
  - (a) Find a basic optimal solution  $x$ . Delete  $v$  such that  $x_v = 0$ . Update each edge  $e \in E(H)$  let  $e \leftarrow e \setminus \{v\}$ . Update matroid  $M \leftarrow M \setminus v$ .
  - (b) For each element  $v$  with  $x_v = 1$ , include  $v$  in  $B$  and decrease  $f(e)$  and  $g(e)$  by 1 for each  $e \ni v$ . Update matroid  $M \leftarrow M/v$ .
  - (c) For every  $e \in E(H)$  such that  $|e| \leq 2\Delta$ , remove  $e$  from  $E(H)$ .
3. Return  $B$ .

**Fig. 1.** The algorithm for the MINIMUM BOUNDED DEGREE MATROID BASIS problem.

also denote by  $M/v$  the matroid obtained by contracting  $v$ , i.e.,  $M/v = (V', \mathcal{I}')$  where  $V' = V \setminus \{v\}$  and  $\mathcal{I}' = \{S \setminus \{v\} : S \in \mathcal{I}, v \in S\}$ .

The algorithm is given in Figure 1. Suppose that the algorithm terminates successfully. Then Theorem 1 follows from a similar argument as in [12], which is sketched as follows. Firstly, observe that the matroid  $M$  is updated to  $M \setminus v$  whenever we remove  $v$  such that  $x_v = 0$  and updated to  $M/v$  whenever we pick  $v$  such that  $x_v = 1$ . A simple verification shows that the *residual* linear programming solution (current LP solution restricted to  $V \setminus \{v\}$ ) remains a feasible solution for the modified linear program in the next iteration. In Step 2c we remove a degree constraint, and hence the current linear programming solution remains a feasible solution. So, a simple inductive argument shows that by only picking elements with  $x_v = 1$ , the cost of the returned basis is no more than the cost of the original basic optimal solution. Also, since we only remove a degree constraint of a hyperedge when it contains at most  $2\Delta$  elements, the degree constraints are violated by at most  $2\Delta - 1$ . Therefore, it remains to show that the algorithm always terminates successfully. That is, it can always find an element  $v$  with  $x_v = 1$  in Step 2b or it finds a hyperedge  $e$  with  $|e| \leq 2\Delta$  in Step 2c.

Suppose for contradiction neither of the above conditions hold. Hence,  $0 < x_v < 1$  for each  $v \in V$  and  $|e| > 2\Delta$  for each  $e \in E(H)$ . Let  $\mathcal{T} = \{S \subseteq V : x(S) = r(S)\}$  be the collection of all tight sets at solution  $x$ . Let  $\chi_S$  denote the characteristic vector of  $S$ , i.e.,  $\chi_S(v) = 1$  if  $v \in S$  else  $\chi_S(v) = 0$ . A family of sets  $\mathcal{L} \subseteq 2^V$  is called a *chain* if the following condition holds: for every  $A, B \in \mathcal{L}$  we have either  $A \subset B$  or  $B \subset A$ . The following claim can be obtained by standard uncrossing argument (see Schrijver [11] Chapter 41).

*Claim.* For any basic solution  $x$ , there exists a chain  $\mathcal{L} \subseteq \mathcal{T}$  such that the following holds.

1.  $\{\chi_S : S \in \mathcal{L}\}$  are linearly independent vectors.
2.  $\text{span}(\{\chi_S : S \in \mathcal{L}\}) = \text{span}(\{\chi_S : S \in \mathcal{T}\})$ .

As  $x$  is a basic solution, there is a set  $E' \subseteq E$  of tight hyperedges (a hyperedge  $e$  is tight if  $x(e) = g(e)$  or  $x(e) = f(e)$ ) such that  $\{\chi_S : S \in \mathcal{L}\} \cup \{\chi_e : e \in E'\}$  are

linearly independent vectors and  $|V| = |E'| + |\mathcal{L}|$ . We now derive a contradiction to this by a counting argument. We assign  $2\Delta$  tokens to each vertex  $v \in V$  for a total of  $2\Delta|V|$  tokens. We then redistribute the tokens so that each hyperedge in  $E'$  collects at least  $2\Delta$  tokens, each member of  $\mathcal{L}$  collects at least  $2\Delta$  tokens, and there are still at least one extra token. This implies that  $2\Delta|V| > 2\Delta|E'| + 2\Delta|\mathcal{L}|$ , which gives us the desired contradiction.

The reassignment is as follows. Each element  $v$  gives  $\Delta$  tokens to the smallest member in  $\mathcal{L}$  it is contained in and one token to each hyperedge  $e \in E'$  it is present in. As any element is contained in at most  $\Delta$  edges, thus the redistribution is valid as we distribute at most  $2\Delta$  tokens per element. Now, consider any set  $S \in \mathcal{L}$  and let  $R$  be the largest set in  $\mathcal{L}$  contained in  $S$ . We have  $x(S) = r(S)$  and  $x(R) = r(R)$ . Thus, we have  $x(S \setminus R) = r(S) - r(R)$ . As constraints for  $R$  and  $S$  are linearly independent and  $x_v > 0$  for each  $v \in V$ , this implies  $r(S) \neq r(R)$ . Since  $r$  is a matroid rank function,  $r(S) - r(R) \geq 1$  as they are both integers. Since  $0 < x_v < 1$ , this implies  $|S \setminus R| \geq 2$ . Thus,  $S$  can collect at least  $2\Delta$  tokens,  $\Delta$  tokens from each element in  $S \setminus R$ , as required. Consider any hyperedge  $e \in E'$ . As  $|e| \geq 2\Delta$  and it can collect one token from each element in  $e$ , there are at least  $2\Delta$  tokens for each edge  $e$ , as required.

Now, it remains to argue that there is an extra token left. If any of the elements is in strictly less than  $\Delta$  hyperedges of  $E'$  then we have one extra token. Else,  $\sum_{e \in E'} \chi_e = \Delta \cdot \chi_V$  which gives dependence among the constraints as  $V \in \mathcal{L}$ . Hence, we have the desired contradiction, and the proof of Theorem 1 follows.  $\square$

Now we show how to use the proof technique of Bansal et al [1] to obtain Theorem 2.

**Proof of Theorem 2:** The proof for upper bounds is similar to the proof of Theorem 1 except for the counting argument. The only important difference is that we remove a hyperedge  $e$  if  $g(e) + \Delta - 1 \geq |e|$ ; this is possible since in that case the degree upper bound on  $e$  can be violated by at most  $\Delta - 1$ . It follows that we may assume that  $|e| - g(e) \geq \Delta$  for all hyperedges.

The proof that  $|V| > |E'| + |\mathcal{L}|$  if  $0 < x < 1$  goes as follows. Let  $\mathcal{L} = \{S_1, \dots, S_k\}$ , where  $S_1 \subsetneq S_2 \subsetneq \dots \subsetneq S_k$ , and let  $S_0 := \emptyset$ . Then  $|e| - x(e) \geq \Delta$  for every  $e \in E'$ , and  $x(S_i \setminus S_{i-1}) = r(S_i) - r(S_{i-1}) \geq 1$  for  $i = 1, \dots, k$ . Using these inequalities, we obtain that

$$\begin{aligned} |E'| + |\mathcal{L}'| &\leq \sum_{e \in E'} \frac{|e| - x(e)}{\Delta} + \sum_{i=1}^k x(S_i \setminus S_{i-1}) \\ &= \sum_{v \in V} \frac{1 - x(v)}{\Delta} |\{e \in E' : v \in e\}| + x(S_k) \leq |V|, \end{aligned}$$

and if equality holds, then  $|\{e \in E' : v \in e\}| = \Delta$  for every  $v \in V$  and  $S_k = V$ . But then  $\Delta \cdot \chi_{S_k} = \sum_{e \in E'} \chi_e$ , which contradicts the linear independence.



If only lower bounds are present, then we can delete a hyperedge  $e$  if  $f(e) \leq \Delta - 1$ , so we may assume that  $f(e) \geq \Delta$  for all hyperedges. To show  $|V| > |E'| + |\mathcal{L}'|$  we use that  $x(e) \geq \Delta$  for every  $e \in E'$  and  $|S_i \setminus S_{i-1}| - x(S_i \setminus S_{i-1}) \geq 1$  for  $i = 1, \dots, k$ , where the latter holds because  $x(S_i \setminus S_{i-1}) < |S_i \setminus S_{i-1}|$  and both are integer. Thus

$$\begin{aligned} |E'| + |\mathcal{L}'| &\leq \sum_{e \in E'} \frac{x(e)}{\Delta} + \sum_{i=1}^k (|S_i \setminus S_{i-1}| - x(S_i \setminus S_{i-1})) \\ &= \sum_{v \in V} \frac{x(v)}{\Delta} |\{e \in E' : v \in e\}| + |S_k| - x(S_k) \leq |V|, \end{aligned}$$

and the claim follows similarly as for upper bounds.  $\square$

*Remark 1.* It is shown in [12] that for the MINIMUM BOUNDED DEGREE SPANNING TREE problem the violation of the degree bounds can be bounded by  $\Delta - 1$  (which is equal to 1 since  $\Delta = 2$  in that problem) even in the presence of both lower and upper bounds on the degrees. In the generalization for matroids, it seems that our method cannot guarantee a solution that violates the bounds by at most  $\Delta - 1$  if both lower and upper degree bounds are present. The reason is that there may be a basic solution with non-integer values, but Step 2c can not be applied, as the following example shows.

Let  $V = \{u_1, u_2, \dots, u_6, v_1, v_2, \dots, v_6\}$  be a ground set of 12 elements, and let  $M = (V, \mathcal{I})$  be the partition matroid where each basis contains 1 element from each of  $\{u_1, v_1\}$ ,  $\{u_3, v_3\}$ ,  $\{u_4, v_2\}$ , and  $\{u_6, v_5\}$ , and 2 elements from  $\{u_2, u_5, v_4, v_6\}$ . Let  $H = (V, E)$  be the hypergraph containing the hyperedges  $\{u_1, u_2, u_3\}$ ,  $\{u_3, u_4, u_5\}$ ,  $\{u_5, u_6, u_1\}$ ,  $\{u_2, u_4, u_6\}$ , and  $\{v_1, v_2, v_3\}$ ,  $\{v_3, v_4, v_5\}$ ,  $\{v_5, v_6, v_1\}$ ,  $\{v_2, v_4, v_6\}$ . For the first four hyperedges, let the lower bound  $f(e)$  be 2, and for the last four hyperedges, let the upper bound  $g(e)$  be 1. Then the following is a basic solution:  $u_i = 2/3$  ( $i = 1, \dots, 6$ ),  $v_i = 1/3$  ( $i = 1, \dots, 6$ ). It is not possible to delete any hyperedges since  $f(e) \geq \Delta$  or  $|e| - g(e) \geq \Delta$  for each hyperedge  $e \in E$ .

## 4 Minimum Bounded Degree Submodular Flow

**Proof of Theorem 3:** The proof of this theorem is also based on the iterative relaxation method used in [10, 12]. Let us define the linear relaxation of the problem by

$$\text{minimize} \quad c(x) = \sum_{e \in E} c(e)x(e) \tag{9}$$

$$x(\delta^{in}(X)) - x(\delta^{out}(X)) \leq b(X) \quad \text{for every } X \subseteq V, \tag{10}$$

$$x(\delta(v)) \geq f(v) \quad \text{for every } v \in V_f, \tag{11}$$

$$x(\delta(v)) \leq g(v) \quad \text{for every } v \in V_g, \tag{12}$$

$$0 \leq x(e) \leq 1 \quad \text{for every } e \in E. \tag{13}$$

Let  $x^*$  be an optimal basic solution of the linear programming relaxation. This can be obtained in polynomial time by the ellipsoid method. Obviously  $c(x^*) \leq \text{OPT}$ . We will find a 0-1 submodular flow of cost at most  $c(x^*)$  that violates the degree bounds by at most one.

The problem can be reduced to an instance containing fewer edges in two cases:

- If  $x^*(e) = 0$  for some  $e \in E$ , then we delete the edge  $e$  from the digraph. A solution of the resulting problem solves the original problem.
- If  $x^*(e) = 1$  for some  $e = uv \in E$ , then we delete the edge  $e$  from the digraph, decrease  $f(u), f(v), g(u), g(v)$  by 1, and change  $b$  as follows:

$$b'(X) = \begin{cases} b(X) - 1 & \text{if } u \notin X \text{ and } v \in X, \\ b(X) + 1 & \text{if } u \in X \text{ and } v \notin X, \\ b(X) & \text{otherwise.} \end{cases}$$

The set function  $b'$  is also crossing submodular. If we have a solution  $x'$  for this modified problem, then we can obtain a solution for the original problem by setting  $x'(e) = 1$ .

This way we can reduce the problem to an instance where  $0 < x^*(e) < 1$  for every  $e \in E$ . We may also delete isolated nodes by changing  $b$  appropriately. Now we try to remove degree bounds so that the solutions of the resulting problem are feasible for the original problem. One difference from the proof of Theorem 1 is that in some iterations we increase the number of vertices in the graph. However, in each step we decrease  $|E| + |V_f| + |V_g|$  by at least one and thus the number of steps is polynomial.

First let us observe that  $g(v) > 0$  for every  $v \in V_g$  and  $f(v) < |\delta(v)|$  for every  $v \in V_f$ , since otherwise there would be some edge  $e$  with  $x^*(e) = 0$  or  $x^*(e) = 1$ . Removal of an upper degree bound at a node  $v$  is possible in the following two cases:

- If  $|\delta(v)| \leq g(v) + 1$ , then we can remove the upper bound at  $v$ , since a solution of the resulting problem cannot violate the original degree bound by more than 1.
- If  $g(v) = 1$ , then we replace  $v$  by two nodes  $v_1$  and  $v_2$ . An edge  $uv \in E$  is replaced by  $uv_1$ , while an edge  $vu \in E$  is replaced by  $v_2u$ . The set function  $b$  is modified as follows:

$$b'(X) = \begin{cases} 1 & \text{if } X = v_1 \text{ or } X = V - v_2, \\ b(X) & \text{if } X \cap \{v_1, v_2\} = \emptyset, \\ b(X - \{v_1, v_2\} + v) & \text{if } \{v_1, v_2\} \subseteq X, \\ \infty & \text{otherwise.} \end{cases}$$

The set function  $b'$  is crossing submodular. No degree upper bound and lower bound are given for  $v_1$  and  $v_2$ , i.e.  $V'_g = V_g - v, V'_f = V_f - v$ . Note that the current solution corresponds to a feasible solution of this relaxation. The

definition of  $b'$  implies that  $x(\delta(v_1)) \leq 1$  and  $x(\delta(v_2)) \leq 1$  for any solution  $x$ . This means that the corresponding solution on the original digraph violates the degree bounds at  $v$  by at most 1.

After the above modifications, we may assume that  $g(v) \geq 2$  and  $|\delta(v)| \geq g(v)+2$  for every  $v \in V_g$ . Removal of a lower degree bound at a node  $v$  is possible in the following two cases:

- If  $f(v) \leq 1$ , then we can remove the lower bound at  $v$ , since a solution of the resulting problem cannot violate the original bound by more than 1.
- If  $f(v) = 2$  and  $|\delta(v)| = 3$ , then we replace  $v$  by two nodes  $v_1$  and  $v_2$ . An edge  $uv \in E$  is replaced by  $uv_1$ , while an edge  $vu \in E$  is replaced by  $v_2u$ . For the modification of  $b$  there are two cases. If  $|\delta^{out}(v)| \leq 1$ , then it is modified as follows:

$$b'(X) = \begin{cases} -1 & \text{if } X = V - v_1, \\ b(X) & \text{if } X \cap \{v_1, v_2\} = \emptyset, \\ b(X - \{v_1, v_2\} + v) & \text{if } \{v_1, v_2\} \subseteq X, \\ \infty & \text{otherwise.} \end{cases}$$

If  $|\delta^{in}(v)| \leq 1$ , then the modified set function is

$$b'(X) = \begin{cases} -1 & \text{if } X = v_2, \\ b(X) & \text{if } X \cap \{v_1, v_2\} = \emptyset, \\ b(X - \{v_1, v_2\} + v) & \text{if } \{v_1, v_2\} \subseteq X, \\ \infty & \text{otherwise.} \end{cases}$$

The set function  $b'$  is crossing submodular. No lower bound is given for  $v_1$  and  $v_2$ , i.e.  $V'_f = V_f - v$ . Note that there is no degree upper bound on  $v$  by the previous rule (since  $g(v) \geq f(v) \geq |\delta(v)| - 1$ ), and the current solution corresponds to a feasible solution in this relaxation. The definition of  $b'$  implies that  $x(\delta(\{v_1, v_2\})) \geq 1$  for any solution  $x$ . This means that the corresponding solution on the original digraph violates the lower bound at  $v$  by at most 1.

After the above modifications, we may assume that  $|\delta(v)| \geq 4$  for every  $v \in V_f \cup V_g$ . The solution corresponding to  $x^*$  is still a feasible solution, but it is not necessarily a basic solution, so we have to resolve the LP and continue this process until a basic solution is obtained where there are no 0-1 edges and no degree bounds can be deleted. (Actually, it is not necessary to solve the LP to optimality, it is enough to perform the easier task of finding a basic solution that is not worse than the current solution.)

At the end of the process either all edges are fixed to 0 or 1 and we are done, or  $0 < x^*(e) < 1$  for every  $e \in E$ , there are no isolated nodes, and  $|\delta(v)| \geq 4$  for every  $v \in V_f \cup V_g$ . We show that the latter case is impossible. Since  $x^*$  is a basic solution, there is a system of linearly independent constraints which are

tight at  $x^*$  for which  $x^*$  is the unique solution of the equation system given by these tight constraints. Let  $\mathcal{F}^*$  be the family of sets corresponding to the submodular flow constraints in this system, and let  $V^*$  denote the set of nodes with degree constraints that are in the system. A family of sets  $\mathcal{F} \subset 2^V$  is called *cross-free* if for every sets  $A, B \in \mathcal{F}$  we have either  $A \subseteq B$ ,  $B \subseteq A$ ,  $A \cap B = \emptyset$  or  $A \cup B = V$ . The following claim can be obtained by standard uncrossing argument (see Schrijver [11] Chapter 60).

*Claim.* We may assume that the family  $\mathcal{F}^*$  is cross-free.

Since  $x^*$  is the unique solution of the equation system defined by  $\mathcal{F}^*$  and  $V^*$ , we have  $|E| \leq |\mathcal{F}^*| + |V^*|$ . We show, using a simple counting argument, that this is impossible.

We assign  $2|E|$  tokens to the nodes by assigning 2 tokens for every edge in  $E$  to the two endpoints of the edge. The idea of the proof is to reassign these tokens to the members of  $\mathcal{F}^*$  and  $V^*$  so that every member gets at least 2 tokens and at least one token is not assigned to any member. This would contradict  $|E| \leq |\mathcal{F}^*| + |V^*|$ .

Let  $r \in V$  be an arbitrary node. We define the family

$$\mathcal{H}^* := \{X \subseteq V - r : X \in \mathcal{F}^*\} \cup \{X \subseteq V - r : V - X \in \mathcal{F}^*\}.$$

Notice that  $\mathcal{H}^*$  is laminar. For a set  $X \in \mathcal{H}^*$ , we define  $X' \in \mathcal{F}^*$  to be either  $X$  or  $V - X$  (depending on which one is in  $\mathcal{F}^*$ ). We will assign 2 tokens to each member of  $\mathcal{H}^*$  so that every member gets tokens from its nodes, thus the tokens of  $r$  are not used.

A node  $v \in V^*$  has at least 4 tokens since  $|\delta(v)| \geq 4$ . We assign 2 of its tokens to  $v$  (as degree constraint) and 2 tokens to the smallest member of  $\mathcal{H}^*$  containing  $v$ . If no member of  $\mathcal{H}^*$  contains  $v$ , we have 2 unused tokens.

To assign tokens to the remaining members of  $\mathcal{H}^*$ , we proceed in an order compatible with the partial order of inclusion. Let  $X \in \mathcal{H}^*$  be a set that has no tokens yet, and let  $\{X_1, \dots, X_k\}$  be the maximal members of  $\mathcal{H}^*$  inside  $X$ , which all have at least two tokens assigned to them. There must be an edge with an endpoint in  $X - \cup_{i=1}^k X_i$ , otherwise the constraints corresponding to  $X', X'_1, \dots, X'_k$  would be linearly dependent: the constraint for  $X'$  would be a  $\pm 1$  combination of the constraints for  $X'_1, \dots, X'_k$ , where the  $i$ -th coefficient depends on whether  $X'_i = X_i$  or  $X'_i = V - X_i$ . Moreover, if only one such edge  $e$  existed, then  $x^*(e)$  would be integer because it would be determined by an integer combination  $b(X'), b(X'_1), \dots, b(X'_k)$ . Since  $0 < x^*(e) < 1$  for every edge, it follows that there are at least two edges with an endpoint in  $X - \cup_{i=1}^k X_i$ , hence there are at least two tokens inside  $X$  that are not yet assigned to other sets. We assign these tokens to  $X$ .

At the end of this procedure, every member of  $\mathcal{H}^*$  and  $V^*$  is assigned 2 tokens, and there is an unused token at  $r$  since it is not an isolated node. This contradicts the assumption that  $|E| \leq |\mathcal{F}^*| + |V^*|$ , so we proved the theorem.  $\square$

#### 4.1 Hardness of The Feasibility Problem

In this section we prove that a special case of the degree-constrained 0-1 submodular flow problem is NP-complete. The construction also shows that the feasibility problems for BOUNDED DEGREE GRAPH ORIENTATION and BOUNDED DEGREE DIRECTED CUT COVER are NP-complete. A subset of edges in a digraph is called *independent* if no two edges have a common node. In the following  $E[W]$  denotes the set of induced edges in  $W$ , i.e. edges with both endpoints in  $W$ .

**Theorem 4.** *Given a digraph  $D = (V, E)$  and a subset  $W \subseteq V$  of nodes, it is NP-complete to decide if it is possible to change the orientation of an independent subset of edges in  $E[W]$  so that the resulting digraph is strongly connected.*

*Proof.* We reduce SAT to this problem. Let us consider a SAT instance with variables  $x_1, \dots, x_n$  and clauses  $c_1, \dots, c_m$ . We associate a digraph  $D = (V, E)$  and a node set  $W \subseteq V$  to this instance using the following construction.

For the variable  $x_j$ , let  $m_j$  be the number of clauses that contain  $x_j$  or  $\neg x_j$ . We construct a cycle of length  $4m_j$ : the nodes are  $u_i^j, v_i^j, w_i^j, z_i^j$  ( $i = 1, \dots, m_j$ ), the oriented edges are  $u_i^j v_i^j, w_i^j v_i^j, z_i^j w_i^j, z_i^j u_{i+1}^j$  ( $i = 1, \dots, m_j$ ). The node set  $W$  consists of all these nodes.

In addition, we add a node  $t$  and nodes  $s_i$  ( $i = 1, \dots, m$ ), and add edges  $s_i t$  ( $i = 1, \dots, m$ ). For a given variable  $x_j$ , suppose that  $c_l$  is the  $l$ -th clause that contains  $x_j$  or  $\neg x_j$ . If it contains  $x_j$ , then we add the edges  $s_l u_l^j, u_l^j s_l, w_l^j t, t w_l^j$ . If it contains  $\neg x_j$ , then we add the edges  $s_l w_l^j, w_l^j s_l, u_l^j t, t u_l^j$ . This finishes the construction of the digraph  $D$ .

Consider the cycle of length  $4m_j$  associated to the variable  $x_j$ . The nodes  $v_i^j$  have out-degree 0, while the nodes  $z_i^j$  have in-degree 0 ( $i = 1, \dots, m_j$ ). This means that we have to change the orientation of  $2m_j$  independent edges in the cycle in order to get a strong orientation. Thus we have two possibilities: either we change the orientation of the edges  $u_i^j v_i^j, z_i^j w_i^j$  ( $i = 1, \dots, m_j$ ), or of the edges  $w_i^j v_i^j, z_i^j u_{i+1}^j$  ( $i = 1, \dots, m_j$ ). We say that the former corresponds to the ‘true’ value of  $x_j$ , while the later corresponds to the ‘false’ value.

In this way, there is a one-to-one correspondence between orientations of the above structure and possible evaluations of the variables. We claim that the orientation is strongly connected if and only if the corresponding evaluation satisfies the SAT formula. Suppose that the formula is not satisfied, i.e. there is a clause  $c_i$  containing only false literals. Consider the node set consisting of  $s_i$  and its neighbors of type  $u$  and  $w$ . By the construction, this set has in-degree 0 in the orientation corresponding to the evaluation. Therefore the orientation cannot be strongly connected.

Now suppose that an evaluation satisfies the formula. Then each node  $s_i$  ( $i = 1, \dots, m$ ) can be reached from  $t$  by a path of length 4 (which corresponds to the “true” literal in  $c_i$ ). Since there is an edge from  $s_i$  to  $t$  for each  $s_i$ , and all other nodes obviously have paths to and from  $t$  or some  $s_i$ , the orientation is strongly connected.  $\square$

**Corollary 6.** *The feasibility problem for degree-constrained 0-1 submodular flows is NP-complete.*

## References

1. N. Bansal, R. Khandekar and V. Nagarajan, *Additive Guarantees for Degree Bounded Directed Network Design*, IBM Research Report RC24347, September 2007.
2. V. Bilo, V. Goyal, R. Ravi and M. Singh, *On the Crossing Spanning Tree Problem*, Proceedings of 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2004: 51–60.
3. K. Chaudhuri, S. Rao, S. Riesenfeld and K. Talwar, *A Push-Relabel Algorithm for Approximating the Minimum-Degree MST Problem and its Generalization to Matroids*, Invited submission to Theoretical Computer Science (Special Issue for ICALP 2006)
4. W.H. Cunningham, *Testing membership in matroid polyhedra*, Journal of Combinatorial Theory, Series B 36(2): 161-188, 1984.
5. J. Edmonds, R. Giles, *A min-max relation for submodular functions on graphs*, Ann. Discrete Math. 1 (1977), 185–204.
6. A. Frank, *An algorithm for submodular functions on graphs*, Ann. Discrete Math. 16 (1982), 97–120.
7. A. Frieze, Personal Communication, March 2007.
8. M.X. Goemans, *Minimum bounded-degree spanning trees*, Proceedings of 47th IEEE FOCS (2006), 273–282.
9. K. Jain, *A factor 2 approximation algorithm for the generalized Steiner network problem*, Combinatorica 21 (2001), 39–60.
10. L.C. Lau, J. Naor, M. Salavatipour and M. Singh, *Survivable network design with degree or order constraints*, Proceedings of 39th ACM STOC (2007), 651–660.
11. A. Schrijver, *Combinatorial Optimization, Polyhedra and Efficiency*, Springer-Verlag 2003.
12. M. Singh and L.C. Lau, *Approximating minimum bounded degree spanning trees to within one of optimal*, In Proceedings of the 39th ACM STOC (2007), 661–670.